Automatic Music Composition with Simple Probabilistic Generative Grammars

Horacio Alberto García Salas, Alexander Gelbukh, Hiram Calvo, and Fernando Galindo Soria

***Abstract*—We propose a model to generate music following a linguistic approach. Musical melodies form the training corpus where each of them is considered a phrase of a language. Implementing an unsupervised technique we infer a grammar of this language. We do not use predefined rules. Music generation is based on music knowledge represented by probabilistic matrices, which we call evolutionary matrices because they are changing constantly, even while they are generating new compositions. We show that the information coded by these matrices can be represented at any time by a probabilistic grammar; however we keep the representation of matrices because they are easier to update, while it is possible to keep separated matrices for generation of different elements of expressivity such as velocity, changes of rhythm, or timbre, adding several elements of expressiveness to the automatically generated compositions. We present the melodies generated by our model to a group of subjects and they ranked our compositions among and sometimes above human composed melodies.**

***Index Terms*—Evolutionary systems, evolutionary matrix, generative grammars, linguistic approach, generative music, affective computing.**

I. Introduction

Music generation does not have a definite solution. We regard this task as the challenge to develop a system to generate a pleasant sequence of notes to human beings and also this system should be capable of generating several kinds of music while resembling human expressivity. In literature, several problems for developing models for fine arts, especially music have been noted. Some of them are: How to evaluate the results of a music generator? How to determine if what such a system produces is music or not? How to say if a music generator system is better than other? Can a machine model expressivity?

Different models have been applied for developing automatic music composers; for example, those based on neural networks [15], genetic algorithms [2, 25] and swarms [4] among other methods.

Manuscript received February 10, 2011. Manuscript accepted for publication July 30, 2011.

H. A. García Salas is with the Natural Language Laboratory, Center for Computing Research, National Polytechnic Institute, CIC-IPN, 07738, DF, México (e-mail: itztzin@gmail.com).

A. Gelbukh was, at the time of submitting this paper, with the Waseda University, Tokyo, Japan, on Sabbatical leave from the Natural Language Laboratory, Center for Computing Research, National Polytechnic Institute, CIC-IPN, 07738, DF, México (e-mail: gelbukh@gelbukh.com).

H. Calvo is with the Natural Language Laboratory, Center for Computing Research, National Polytechnic Institute, CIC-IPN, 07738, DF, México (e-mail: hcalvo@cic.ipn.mx).

F. Galindo Soria is with Informatics Development Network, REDI (e-mail: fgalindo@ipn.mx).

In order to generate music automatically we developed a model that describes music by means of a linguistic approach; each musical composition is considered a phrase that is used to learn the musical language by inferring its grammar. We use a learning algorithm that extracts musical features and forms probabilistic rules that afterwards are used by a note generator algorithm to compose music. We propose a method to generate linguistic rules [24] finding musical patterns on human music compositions. These patterns consist of sequences of notes that characterize a melody, an author, a style or a music genre. The likelihood of these patterns of being part of a musical work is used by our algorithm to generate a new musical composition.

To model the process of musical composition we rely on the concept of *evolutionary systems* [8], in the sense that systems evolve as a result of constant change caused by flow of matter, energy and information [10]. Genetic algorithms, evolutionary neural networks, evolutionary grammars, evolutionary cellular automata, evolutionary matrices, and others are examples of evolutionary systems. In this work we follow the approach of evolutionary matrices [11].

This paper is organized as follows. In Section II we present works related to automatic music composition. In Section III, we describe our model. In Section IV, we describe an algorithm to transform a matrix into a grammar. In Section V we show how we handle expressivity in our model. In Section VI, we present results of a test to evaluate generated music. Finally, in Section VII, we present some conclusions of our model and future work to improve our model.

II. Related Work

A. Review Stage

An outcome of development of computational models applied to humanistic branches as fine arts like music is generative music or music generated from algorithms.

Different methods have been used to develop music composition systems, for example: noise [5], cellular automata [20], grammars [13, 22], evolutionary methods [13], fractals [14, 16], genetic algorithms [1], case based reasoning [19], agents [21] and neural networks [7, 15]. Some systems are called hybrid since they combine some of these techniques. For a comprehensive study please refer to [23] and [17].

Harmonet [15] is a system based on connectionist networks, which has been trained to produce chorale style of J. S. Bach. It focuses on the essence of musical information, rather than restrictions on music structure. Eck and Shmidhuber [7] believe that music composed by recurrent neural networks lacks structure, and do not maintain memory of distant events.

They developed a model based on LSTM (Long Short Term Memory) to represent the overall and local music structure, generating blues compositions.

Kosina [18] describes a system for automatic music genre recognition based on audio content signal, focusing on musical compositions of three music genres: classical, metal and dance. Blackburm and DeRoure [3] present a system to recognize through the contents of a music database, with the idea to make search based on music contours, i.e. in a relative changes representation in a musical composition frequencies, regardless of tone or time.

There is a number of works based on evolutionary ideas for music composition. For example, Ortega *et al.* [22] used generative context-free grammars for modeling the musical composition. Implementing genetic algorithms they made grammar evolve to improve the musical generation. GenJam [1] is a system based on a genetic algorithm that models a novice jazz musician learning to improvise. It depends on user feedback to improve new compositions through several generations.

Todd and Werner [25] developed a genetic algorithm based on co-evolution, learning and rules. In their music composer system there are male individuals that produce music and female critics that evaluate it to mate them. After several generations they create new musical compositions.

In our approach we focus on the following points:

– The evolutionary aspect—to keep learning while generating;
– Stressing the linguistic metaphor of musical phrases and textual phrases, words and sets of notes;
– Adding expressiveness to achieve a more human aspect;
– Studying the equivalence between a subset of grammar rules and matrices [11].

III. Music Generation

A musical composition is a structure of note sequences made of other structures built over time. How many times a musical note is used after another reflects patterns of sequences of notes that characterizes a genre, style or an author of a musical composition. We focus on finding patterns on monophonic music.

A. Linguistic approach

Our model is based on a linguistic approach [9]. We describe musical compositions as phrases made up of sequences of notes as lexical items that represent sounds and silences throughout time. The set of all musical compositions forms the musical language.

In the following paragraphs we define some basic concepts that we will use in the rest of this paper.

Definition 1: A *note* is a representation of tone and duration of musical sound.

Definition 2: The *alphabet* is the set of all notes: alphabet = {notes}.

Definition 3: A *musical composition m* is an arrangement of musical notes: Musical composition = $a_1\ a_2\ a_3 \ldots a_n$ where $a_i \in$ {notes}.

In our research we work with musical compositions *m* of monophonic melodies, modeling two variables of notes: musical frequencies and musical tempos. We split these variables to form a sequence of symbols with each of them.

Definition 4: The *Musical Language* is the set of all musical compositions: Musical Language = {musical compositions}.

For example, having the sequence of notes (frequencies) of musical composition "*El cóndor pasa*" (the condor passes by): $b\ e\ d_{\#}\ e\ f_{\#}\ g\ f_{\#}\ g\ a\ b_2\ d_2\ b_2\ e_2\ d_2\ b_2\ a\ g\ e\ g\ e\ b\ e\ d_{\#}\ e\ f_{\#}\ g\ f_{\#}\ g\ a\ b_2\ d_2\ b_2\ e_2\ d_2\ b_2\ a\ g\ e\ g\ e\ b_2\ e_2\ d_2\ e_2\ d_2\ e_2\ g_2\ e_2\ d_2\ e_2\ d_2\ b_2\ g\ e_2\ d_2\ e_2\ d_2\ e_2\ g_2\ e_2\ d_2\ e_2\ d_2\ b_2\ a\ g\ e\ g\ e$

We assume this sequence is a phrase of musical language.

B. Musical Evolutionary System

Evolutionary systems interact with their environment finding rules to describe phenomena and use functions that allow them to learn and adapt to changes. A scheme of our evolutionary model is shown in Fig. 1.

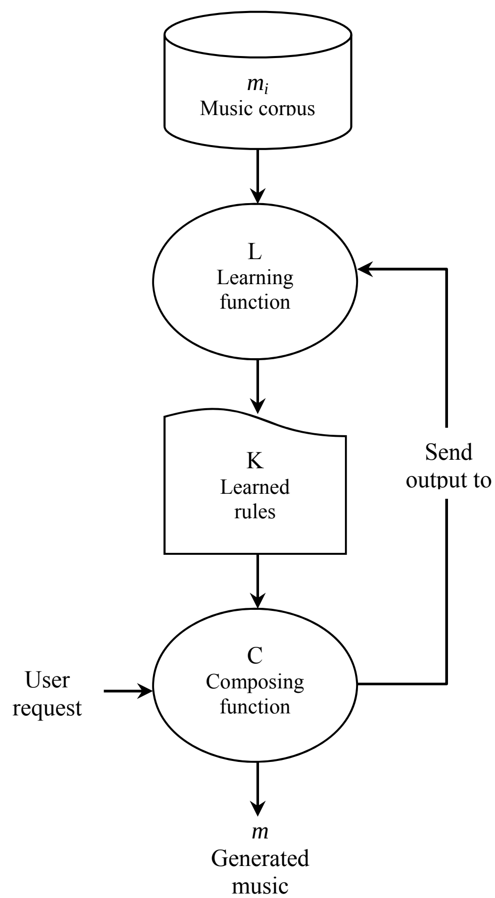

Fig. 1. Model.

The *workspace* of musical language rules is represented by K and there exist many ways to make this representation, e.g. grammars, matrices, neural nets, swarms and others. Each musical genre, style and author has its own rules of composition. Not all of these rules are described in music theory. To make automatic music composition we use an evolutionary system to find rules *K* in an unsupervised way.

The function L is a learning process that generates rules from each musical composition m_i creating a representation of musical knowledge. The evolutionary system originally does not have any rule. We call K_0 when K is empty. While new musical examples m_0, m_1, …, m_i are learned K is modified from K_0 to K_{i+1}.

$$L(m_i, K_i) = K_{i+1}$$

Function L extracts musical features of m_i and integrates them to K_i generating a new representation K_{i+1}. This makes knowledge representation K evolves according to the learned examples.

These learned rules K are used to generate musical composition *m* automatically. It is possible to construct a function C(K) where C is called musical composer. Function C uses K to produce a novel musical composition *m*.

$$C(K) = m$$

For listening of the new music composition there is a function I called musical interpreter or performer that generates the sound.

$$I(m) = \text{sound}$$

Function I takes music *m* generated by function C to stream it to the sound device. We will not discuss this function in this paper.

C. Learning Module based on Evolutionary Matrices

To describe our music learning module we need to define several concepts. Let L be a learning process as the function that extracts musical features and adds this information into K. There are different ways to represent K. In our work we use a matrix representation. We will show in Section IV that this is equivalent to a probabilistic grammar.

Definition 5: Musical Frequency = {musical frequencies} where musical frequencies (*mf*) are the number of vibrations per second (Hz) of notes.

Definition 6: Musical Time = {*musical times*} where *musical times* (mt) are durations of notes.

Function L receives musical compositions *m*. Musical Composition $m = a_1\, a_2\, a_3 \ldots a_n$ where $a_i=\{f_i,t_i\}$, $i \in [1,n]$, $f_i \in$ Musical Frequency, $t_i \in$ Musical Time, $[1,n] \subset \mathrm{N}$

To represent rules K we use matrices for musical frequencies and for musical times. We refer to them as rules M. Originally these matrices are empty; they are modified with every musical example.

Rules M are divided by function L into MF and MT where MF is the component of musical frequencies (*mf*) rules extracted from musical compositions and MT is the component of musical time (*mt*) rules.

We are going to explain how L works with musical frequency matrix MF. Time matrix MT works the same way.

Definition 7: MF is a workspace formed by two matrices. One of them is a *frequency distribution matrix* (FDM) and the other one is a *cumulative frequency distribution matrix* (CFM).

Each time a musical composition m_i arrives, L upgrades FDM. Then it recalculates CFM, as follows:

Definition 8: Let *FrequencyNotes* be an array in which are stored the numbers corresponding to a musical composition notes.

Definition 9: Let *n* be the number of notes recognized by the system, $n \in \mathrm{N}$.

Definition 10: Frequency Distribution Matrix (FDM) is a matrix with *n* rows and *n* columns.

Given the musical composition $m = f_1\, f_2\, f_3 \ldots f_r$ where $f_i \in$ *FrequencyNotes*. The learning algorithm of L to generate the frequency distribution matrix FDM is:

$$\forall i \in [1,r], [1,r] \subset \mathrm{N}, \mathrm{FDM}_{fi,fi+1} = \mathrm{FDM}_{fi,fi+1}+1,$$
where $\mathrm{FDM}_{fi,fi+1} \in \mathrm{FDM}$.

Definition 11: Cumulative Frequency Distribution Matrix CFM is a matrix with *n* rows and *n* columns.

The algorithm of L to generate cumulative frequency distribution matrix CFM is:

$$\forall i \in [1,n], \forall j \in [1,n], [1,n] \subset \mathrm{N}, \forall\, \mathrm{FDM}_{i,j} \neq 0$$

$$\mathrm{CFM}_{i,j} = \sum_{k=1}^{j} FDM_{i,k}$$

These algorithms to generate MF, the workspace formed by FDM and CFM, are used by function L with every musical composition m_i. This makes the system evolve recursively according to musical compositions m_0, m_1, m_2,…, m_i.

$$L(m_i, \ldots L(m_2, L(m_1, L(m_0, MF_0)))) = MF_{i+1}$$

D. Composer Function C: Music Generator Module

Monophonic music composition is the art of creating a single melodic line with no accompaniment. To compose a melody a human composer uses his/her creativity and musical knowledge. In our model composer function C generates a melodic line based on knowledge represented by cumulative frequency distribution matrix CFM.

For music generation is necessary to choose next note. In our model each *i* row of CFM represents a probability function for each *i* note on which is based the decision of the next note. Each *j* column different of zero represents possible notes to follow the *i* note. The most probable notes form characteristic musical patterns.

Definition 12: T_i and T.
Let T_i to be an element where it is store the total of cumulative frequency sum of each *i* row of FDM.

$$\forall i \in [1,n], [1,n] \subset \mathrm{N},\ \mathrm{T}_i = \sum_{k=1}^{n} FDM_{i,k}$$

Let T be a column with *n* elements where it is store the total of cumulative frequency sum of FDM.

Note generation algorithm:

```
while(not end)
{
  p=random(Ti)
  while(CFMi,j < p)
    j=j+1
  next note=j
  i=j
}
```

E. Example

Let us take the sequence of frequencies of musical composition “*El cóndor pasa*”:

b e $d_\#$ e $f_\#$ g $f_\#$ g a b_2 d_2 b_2 e_2 d_2 b_2 a g e g e b e $d_\#$ e $f_\#$ g $f_\#$ g a b_2 d_2 b_2 e_2 d_2 b_2 a g e g e b_2 e_2 d_2 e_2 d_2 e_2 g_2 e_2 d_2 e_2 d_2 b_2 g e_2 d_2 e_2 d_2 e_2 g_2 e_2 d_2 e_2 d_2 b_2 a g e g e

FrequencyNotes = {**b**, $\mathbf{d_\#}$, **e**, $\mathbf{f_\#}$, **g**, **a**, $\mathbf{b_2}$, $\mathbf{d_2}$, $\mathbf{e_2}$, $\mathbf{g_2}$} are the terminal symbols or alphabet of this musical composition. They are used to tag each row and column of frequency distribution matrix FDM. Each number stored in FDM of Fig. 2, represents how many times a row note was followed by a column note in *condor pasa* melody. To store the first note of each musical composition S row is added, it represents the axiom or initial symbol. Applying the learning algorithm of L we generate frequency distribution matrix FDM of Fig. 2.

	b	$d_\#$	e	$f_\#$	g	a	b_2	d_2	e_2	g_2
S	1									
b			2							
$d_\#$			2							
e	1	2		2	3		1			
$f_\#$					4					
g			6	2		2			1	
a					3		2			
b_2					1	3		2	3	
d_2							6		6	
e_2								10		2
g_2									2	

Fig. 2. Frequency distribution matrix FDM.

We apply the algorithm of L to calculate cumulative frequency distribution matrix CFM of Fig. 3 from frequency distribution matrix FDM of Fig. 2. Then we calculate each T_i of T column.

For generation of a musical composition we use note generator algorithm. Music generation begins by choosing the first composition note. S row of matrix of Fig. 3 contains all possible beginning notes. In our example only the **b** note can be chosen. Then **b** is the first note and the *i* row of $\mathrm{CFM}_{i,j}$ which we use to determine second note. Only the **e** note can be chosen after the first note **b**.

So the first two notes of this new musical melody are m_{i+1}={**b**, **e**}. Applying note generator algorithm to determine third note: We take the value of column $\mathrm{T_e}$=9. A *p* random number between zero and 9 is generated, suppose *p*=6. To find next note we compare *p* random number with each non-zero value of **e** row until one greater than or equal to this number is found. Then column **g** is the next note since $\mathrm{M_{e,g}}$=8 is greater than *p* = 6. The column *j* = **g** is where it is stored this number that indicates the following composition note and the following *i* row to be processed. The third note of new musical composition m_{i+1} is **g**. So m_{i+1} = {**b**, **e**, **g**,…}. Then to determine the fourth note we must apply the note generator algorithm to *i* = **g** row.

Since each non-zero value of *i* row represents notes that used to follow *i* note, then we will generate patterns according to probabilities learned from musical compositions examples.

	b	$d_\#$	e	$f_\#$	g	a	b_2	d_2	e_2	g_2	T
S	1										1
b			2								2
$d_\#$			2								2
e	1	3		5	8		9				9
$f_\#$					4						4
g			6	8		10			11		11
a					3		5				5
b_2					1	4		6	9		9
d_2							6		12		12
e_2								10		12	12
g_2									2		2

Fig. 3. Cumulative frequency distribution matrix CFM.

IV. Matrices and Grammar

Our work is based on a linguistic approach and we have used a workspace represented by matrices to manipulate music information. Now we show that this information representation is equivalent to a probabilistic generative grammar.

There are different ways to obtain a generative grammar G,. From frequency distribution matrix FDM and total column T, it is possible to construct a probabilistic generative grammar.

Definition 13: MG is a workspace formed by FDM and a probabilistic grammar G.

To generate a grammar first we generate a probability matrix PM determined from frequency distribution matrix FDM.

Definition 14: Probability Matrix (PM) is a matrix with *n* rows and *n* columns.

The algorithm to generate probability matrix PM is:

$\forall i \in [1,n], \forall j \in [1,n], \forall \mathrm{FDM}_{i,j} \neq 0 \qquad \mathrm{PM}_{i,j} = \mathrm{FDM}_{i,j}/\mathrm{T}_i$

There is a probabilistic generative grammar G{$\mathrm{V_n}$, $\mathrm{V_t}$, S, P, Pr} such that G can be generated from PM. $\mathrm{V_n}$ is the set of nonterminals symbols, $\mathrm{V_t}$ is the set of all terminal symbols or alphabet which represents musical composition notes. S is the axiom or initial symbol, P is the set of rules generated and Pr is the set of rules probabilities represented by values of matrix PM.

For transforming the PM matrix in a grammar we use the following algorithm:

1. Build the auxiliary matrix AM from PM:
    a. substitute each row i tag of PM with a nonterminal symbol X_i except S row which is copied as it is
    b. substitute each column j tag by its note f_j and a nonterminal symbol X_j
    c. copy all values of cells of matrix PM into corresponding cells of matrix AM
2. For each row i and each column j such that $AM_{i,j} \neq 0$
    a. i row corresponds to grammar rule X_i
    b. j column corresponds to a terminal symbol f_j and a nonterminal symbol X_j with probability $p_{i,j}$

Then rules of grammar G are of the form $X_i \rightarrow f_j\ X_j\ (p_{i,j})$. This is a grammatical representation of our model. For each music composition m_i a MG, the workspace formed by FDM and grammar G, can be recursively generated.

$$L(m_i, \ldots L(m_2, L(m_1, L(m_0, MG_0))))=MG_{i+1}$$

A. Example

From frequency distribution matrix FDM of Fig. 2 it is generated probability matrix PM of Fig. 4.

	b	$d_\#$	e	$f_\#$	g	a	b_2	d_2	e_2	g_2
S	1									
b			1							
$d_\#$			1							
e	1/9	2/9		2/9	3/9		1/9			
$f_\#$					1					
g			6/11	2/11		2/11			1/11	
a					3/5		2/5			
b_2					1/9	3/9		2/9	3/9	
d_2							6/12		6/12	
e_2								10/12		2/12
g_2									1	

Fig. 4. Probability matrix PM.

	bX_1	$d_\#X_2$	eX_3	$f_\#X_4$	gX_5	aX_6	b_2X_7	d_2X_8	e_2X_9	g_2X_{10}
S	1									
X_1			1							
X_2			1							
X_3	1/9	2/9		2/9	3/9		1/9			
X_4					1					
X_5			6/11	2/11		2/11			1/11	
X_6					3/5		2/5			
X_7					1/9	3/9		2/9	3/9	
X_8							6/12		6/12	
X_9								10/12		2/12
X_{10}									1	

Fig. 5. Auxiliary matrix AM.

From matrix PM of Fig. 4 the auxiliary matrix AM of Fig. 5 is generated. From given AM matrix of Fig. 5 We can generate grammar $G\{V_n, V_t, S, P, Pr\}$. Where $V_n=\{S, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}\}$ is the set of non-terminals symbols. $V_t=\{b, d_\#, e, f_\#, g, a, b_2, d_2, e_2, g_2\}$ is the set of all terminal symbols or alphabet. S is the axiom or initial symbol. Pr is the set of rules probabilities represented by values of matrix AM. Rules P are listed in Fig. 6.

$S \rightarrow b\ X_1(1)$
$X_1 \rightarrow e\ X_3(1)$
$X_2 \rightarrow e\ X_3(1)$
$X_3 \rightarrow b\ X_1(1/9) \mid d_\#\ X_2(2/9) \mid f_\#\ X_4(2/9) \mid g\ X_5(3/9) \mid b_2\ X_7(1/9)$
$X_4 \rightarrow g\ X_5(1)$
$X_5 \rightarrow e\ X_3(6/11) \mid f_\#\ X_4(2/11) \mid a\ X_6(2/11) \mid e_2\ X_9(1/11)$
$X_6 \rightarrow g\ X_5(3/5) \mid b_2\ X_7(2/5)$
$X_7 \rightarrow g\ X_5(1/9) \mid a\ X_6(3/9) \mid d_2\ X_8(2/9) \mid e_2\ X_9(3/9)$
$X_8 \rightarrow b_2\ X_7(6/12) \mid g_2\ X_{10}(6/12)$
$X_9 \rightarrow d_2\ X_8(10/12) \mid g_2\ X_{10}(2/12)$
$X_{10} \rightarrow e_2\ X_9(1)$

Fig. 6. Probabilistic generative grammar.

V. Expressivity

Expressivity can be regarded as a mechanism that displays transmission and interpretation vividness of feelings and emotions. For example fear in front of a threat. Physical factors interfere like cardiac rhythm, changes in respiratory system, in endocrine system, in muscular system, in circulatory system, secretion of neurotransmitters, etc. Another important factor is empathy which is the capacity of feelings and emotions recognition in others [6]. It is out of our research to explain how these physical changes are made or how empathy takes place among living beings. We just simulate expressivity in music generation.

A. Expressivity within our Model

Music can be broken down into different functions that characterize it like frequency, time and intensity. So each note of a melody is a symbol with several features or semantic descriptors that give the meaning of a long or short sound, low, high, intense, soft, of a guitar or of a piano.

With our model is possible to represent each of these variables using matrices or grammars that reflect their probabilistic behavior. In this paper we have presented how to model frequency and time. We can build an intensity matrix the same way. With more variables more expressivity the generated music will reflect.

Using our model we can characterize different kinds of music based on its expressivity, for example in happy music or sad music. Besides we have the possibility of mixing features of distinct kinds of music, for example frequency functions of happy music with time functions of sad music. Also we can combine different genres like classic times with rock frequencies. So in addition of generating music we can invent new genres and music styles.

VI. Results

In order to evaluate whether our algorithm is generating music or not, we decided to conduct a Turing-like test. Participants of this test had to tell us if they like music generated by our model, without them knowing that it was automatically music generated. This way we sought the answer to two questions: whether or not we are doing music and whether or not our music is pleasant.

We compiled 10 melodies, 5 of them generated by our model and another 5 by human composers and we asked human subjects to rank melodies according to whether they liked them or not, with numbers between 1 and 10 being number 1 the most they liked. None of subjects knew about the order of music compositions. These 10 melodies were presented as in Table I.

TABLE I
ORDER OF MELODIES AS THEY WERE PRESENTED TO SUBJECTS

ID	Melody	Author
A	*Zanya*	(generated)
B	*Fell*	Nathan Fake
C	*Alucin*	(generated)
D	*Idiot*	James Holden
E	*Ciclos*	(generated)
F	*Dali*	Astrix
G	*Ritual Cibernetico*	(generated)
H	*Feelin' Electro*	Rob Mooney
I	*Infinito*	(generated)
J	*Lost Town*	Kraftwerk

We presented this test to more than 30 participants in different places and events. We sought that the characteristics of these participants were as varied as possible (age, gender and education), however most of them come from a related IT background. Test results were encouraging, since automatically generated melodies were ranked at 3rd and 4th place above human compositions. Table II shows the ranking of melodies as a result of the Turing-like test we developed.

TABLE II
ORDER OF MELODIES OBTAINED AFTER THE TURING-LIKE TEST

ID	Ranking	Melody	Author
B	1	*Fell*	Nathan Fake
D	2	*Idiot*	James Holden
C	3	*Alucín*	(generated)
A	4	*Zanya*	(generated)
F	5	*Dali*	Astrix
H	6	*Feelin' Electro*	Rob Mooney
J	7	*Lost Town*	Kraftwerk
E	8	*Ciclos*	(generated)
G	9	*Ritual Cibernético*	(generated)
I	10	*Infinito*	(generated)

VII. CONCLUSIONS AND FUTURE WORK

We proposed an evolutionary model based on evolutionary matrices for musical composition. Our model is learning constantly, increasing its knowledge for generating music while more data is presented. It does not need any predefined rules. It generates them from phrases of the seen language (musical compositions) in an unsupervised way.

As we shown, our matrices can be expressed as probabilistic grammar rules, so that we can say that our systems extracts grammar rules dynamically from musical compositions. These rules generate a musical language based on the compositions presented to the system. These rules can be used to generate different musical phrases, meaning new musical compositions. Because the probabilistic grammars learned can generalize a language beyond the seen examples of it, our model has what can be called innovation, which is what we are looking for music creation, while keeping the patterns learned from human music.

As a short-term future work we plan to characterize different kinds of music, from sad to happy, or from classic to electronic in order to find functions for generating this kind of music. We are also developing the use of other matrices to consider more variables involved in a musical work, such as velocity, fine-graded tempo changes, etc., thus adding more expressivity to the music created by our model.

ACKNOWLEDGEMENTS

The work was done under partial support of Mexican Government (CONACYT 50206-H, SIP-IPN 20113295, COFAA-IPN, PIFI-IPN, SNI).

REFERENCES

[1] J. A. Biles, “GenJam: Evolution of a Jazz Improviser,” *Creative Evolutionary Systems, Section: Evolutionary Music*, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., 2001, pp. 165–187.

[2] D. Birchfield, “Generative Model for the Creation of Musical Emotion, Meaning and Form,” in *Proceedings of the 2003 International Multimedia Conference ACM SIGMM*, Berkeley, California: Workshop on Experiential Telepresence, Session: Playing experience, 2003, pp. 99–104.

[3] S. Blackburm, and D. DeRoure, “A tool for content based navigation of music,” in *Source International Multimedia Conference. Proceedings of the sixth ACM international conference on Multimedia*. Bristol, United Kingdom, 1998, pp. 361–368.

[4] T. Blackwell, “Swarming and Music,” *Evolutionary Computer Music*. Springer London, 2007, pp. 194–217.

[5] M. Bulmer, “Music From Fractal Noise,” in *Proceedings of the Mathematics 2000 Festival*, University of Queensland, Melbourne, 2000.

[6] T. Cochrane, “A Simulation Theory of Musical Expressivity,” *The Australasian Journal of Philosophy*, Volume 88, Issue 2, 191–207, 2010.

[7] D. Eck, and J. Schmidhuber, *A First Look at Music Composition using LSTM Recurrent Neural Networks*, Source Technical Report: IDSIA-07-02. Publisher Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 2002.

[8] F. Galindo Soria, “Sistemas Evolutivos: Nuevo Paradigma de la Informática,” en *Memorias XVII Conferencia Latinoamericana de Informática*, Caracas Venezuela, 1991.

[9] F. Galindo Soria, “Enfoque Lingüístico,” en *Memorias del Simposio Internacional de Computación de 1995*, Cd. de México: Instituto Politécnico Nacional CENAC, 1995.

[10] F. Galindo Soria, *Teoría y Práctica de los Sistemas Evolutivos*, Cd. de México, 1997.

[11] F. Galindo Soria, “Matrices Evolutivas,” en *Memorias de la Cuarta Conferencia de Ingeniería Eléctrica CIE/98,* Cd. de México: Instituto Politécnico Nacional, CINVESTAV, 1998, pp. 17–22.

[12] A. García Salas, *Aplicación de los Sistemas Evolutivos a la Composición Musical*, México D.F: Tesis de maestría, Instituto Politécnico Nacional UPIICSA, 1998.

[13] A. García Salas, A. Gelbukh, and H. Calvo, “*Music Composition Based on Linguistic Approach*,” in *Proceedings of the 9th Mexican International Conference on Artificial Intelligence*, Pachuca, México, 2010, pp. 117–128.

[14] M. Gardner, “Mathematical Games: White and Brown Music, Fractal Curves and One-Over-f Fluctuations,” *Scientific American*, 4, 16–32, 1978.

[15] H. Hild, J. Feulner, and W. Menzel, “Harmonet: A Neural Net for Harmonizing Chorales in the Style of J. S. Bach,” *Neural Information Processing 4*. Germany: Morgan Kaufmann Publishers Inc., 1992, pp. 267–274.

[16] R. Hinojosa, *Realtime Algorithmic Music Systems From Fractals and Chaotic Functions: Toward an Active Musical Instrument*, Barcelona: PhD Thesis, Universitat Pompeu Fabra, 2003.

[17] H. Järveläinen, "Algorithmic Musical Composition," in *Seminar on content creation Art@Science*, Helsinki: University of Technology, Laboratory of Acoustics and Audio Signal Processing, 2000.

[18] K. Kosina, "Music Genre Recognition." *Diplomarbeit. Eingereicht am Fachhochschul-Studiengang. Mediente Chnik Und Design in Hagenberg*, 2002.

[19] G. Maarten, J.L. Arcos, and R. López de Mántaras, "A Case Based Approach to Expressivity-Aware Tempo Transformation," *Machine Learning*, 65(2-3): 411–437, 2006.

[20] K. McAlpine, E. Miranda, and S. Hoggar, "Making Music with Algorithms: A Case-Study System," *Computer Music Journal*, 23(2): 19–30, 1999.

[21] M. Minsky, "Music, Mind, and Meaning." *Computer Music Journal*, 5(3), 1981.

[22] A P. Ortega, A.R. Sánchez, and M. M. Alfonseca, "Automatic composition of music by means of Grammatical Evolution," *ACM SIGAPL APL*, 32(4): 148–155, 2002.

[23] G. Papadopoulos, and G. Wiggins, "AI Methods for Algorithmic Composition: A Survey, a Critical View and Future Prospects," in *Symposium on Musical Creativity 1999,* University of Edinburgh, School of Artificial Intelligence Division of Informatics, 1999, pp. 110–117.

[24] Y. Ledeneva and G. Sidorov, "Recent Advances in Computational Linguistics," *Informatica. International Journal of Computing and Informatics*, 34, 3–18, 2010.

[25] P.M. Todd and G.M. Werner, "Frankensteinian Methods for Evolutionary Music Composition," *Musical networks: Parallel distributed perception and performance*, MA, USA: Cambridge, MIT, Press Bradford Books, 1999.